



Таблица 2

Анализ сайтов по критериям

Сайт	Дизайн	Юзабилити	Стиль текста	Содержание	Верстка	SEO	Активность	Посещаемость	Время загрузки	Реклама	Среднее значение
Samgups.ru	6	5	7	8	4	2	9	6	3	1	5,1
Samgtu.ru	10	7	10	9	9	8	10	7	8	0	7,8

Таким образом, мы наглядно видим, что у сайта Samgups.ru серьезные проблемы с SEO оптимизацией и временем загрузки, а у сайта Samgtu.ru практически все находится на высоком уровне.

**Выводы.** Выявлены достоинства и недостатки CMS платформ. Составлены основные 10 критериев, на основе которых можно проводить оценку сайтов. Представлен пример применения данной работы на практике, который наглядно показывает все плюсы и минусы анализируемых сайтов.

### Литература

1. Рамел Д. Самоучитель Joomla! // Знакомство с Joomla!. 2008. С. 15-25.
2. Ромашов В.Р. CMSDrupal система управления содержимым сайта // Установка и использование Drupal. 2010. С. 8-15.
3. Грачев А. Создай свой сайт на WordPress // Знакомство с CMSWordPress. 2011. С. 14-23.
4. Рамел Д. Самоучитель Joomla! // Статистика сайта. 2008. С. 341-355.

А.В. Баландин, Р.А. Кавков

### ЭВОЛЮЦИОННОЕ ПРОТОТИПИРОВАНИЕ РАСПРЕДЕЛЁННЫХ ПРИЛОЖЕНИЙ РЕАЛЬНОГО ВРЕМЕНИ В ОСРВ QNX NEUTRINO

(Самарский университет)

Приложения реального времени (ПРВ) составляют особый класс программ, используемых в системах промышленной автоматизации, системах автоматизации испытаний и научных исследований, робототехнических системах, системах машинного зрения, системах аварийной сигнализации и т.п. На ПРВ возлагают функции оперативного контроля и управления системными процессами в режиме реального времени, что в значительной степени определяет эффективность и надёжность функционирования таких систем в течение всего жизненного цикла. Поэтому априорный анализ способности создаваемого ПРВ



удовлетворять заданным требованиям оперативного функционирования начинается на самых ранних стадиях разработки и продолжается в течение всего периода разработки. Этим обуславливается то, что эволюционное прототипирование (когда последовательно масштабируемый, модифицируемый и периодически тестируемый натурный макет поэтапно приближается к реальному приложению) является единственным надёжным способом разработки ПРВ с заданными свойствами.

Существует множество коммерческих средств программирования ПРВ (SCADA-систем), явно или неявно поддерживающих технологию эволюционного прототипирования. Например, WinCC OA, Trace mode, LabVIEW и множество других им подобных. Однако их приобретение для «одноразового» программирования «не сложных» систем не всегда оправдано.

В качестве альтернативы коммерческим средствам предлагается технология эволюционного прототипирования ПРВ, базирующаяся разработанном методе структуризации ПРВ [1, 2] и использовании стандартных средств программирования, предоставляемых современными POSIX-ориентированными операционными системами реального времени (ОСРВ), которыми обычно оснащаются вычислительные системы, используемые ПРВ. Предполагается, что на ПРВ возлагается сбор информации, её анализ и выработка управляющих воздействий в реальном времени, а также хранение, отображение и передача информации в локальной сети (ЛС). В данной работе в качестве ОСРВ рассматривается QNX Neutrino [3].

Прототип ПРВ формируется на четырёх уровнях:

1. Агрегатном.
2. Процессном.
3. Нитевом.
4. Процедурном.

На рисунке 1 приведён пример представления структуры абстрактного ПРВ на *агрегатном уровне* как параллельно и асинхронно функционирующие в узлах ЛС *программные агрегаты*, взаимодействующие друг с другом посредством однонаправленной передачи сообщений. Узлы ЛС имеют символические имена:  $Comp1, \dots, Comp4$ . В каждом узле располагается по одному агрегату:  $A_1, \dots, A_4$ . Взаимодействие агрегатов между собой осуществляется по однонаправленным каналам передачи сообщений:  $c_1, c_2, \dots, c_9$ . Агрегаты могут взаимодействовать и с внешними устройствами узлов ЛС (контроллеры, внешняя память, устройства отображения, пульт оператора).

Универсальная структура программного агрегата на *процессном уровне* представлена на рисунке 2. Агрегат состоит из *базы оперативных данных* (БОД) и *операционных модулей* преобразования данных. БОД осуществляет сбор исходных оперативных данных и временное хранение всех данных в именованных «абстрактных ячейках» оперативной памяти:  $m_1, m_2, \dots, m_m$ . Оперативные данные являются *темпоральными данными* трёх типов: *датумы*, *импульсы*, *моды* [3]. Для управления оперативными данными в состав БОД входят специальные модули. *Модули генерации данных* являются поставщиками исходных



темпоральных данных в ячейки памяти БОД. Они взаимодействуют с *окружением* агрегата, обеспечивающим модулям связь с внешними устройствами узла ЛС, например контроллером, пультом оператора или каналами передачи данных.

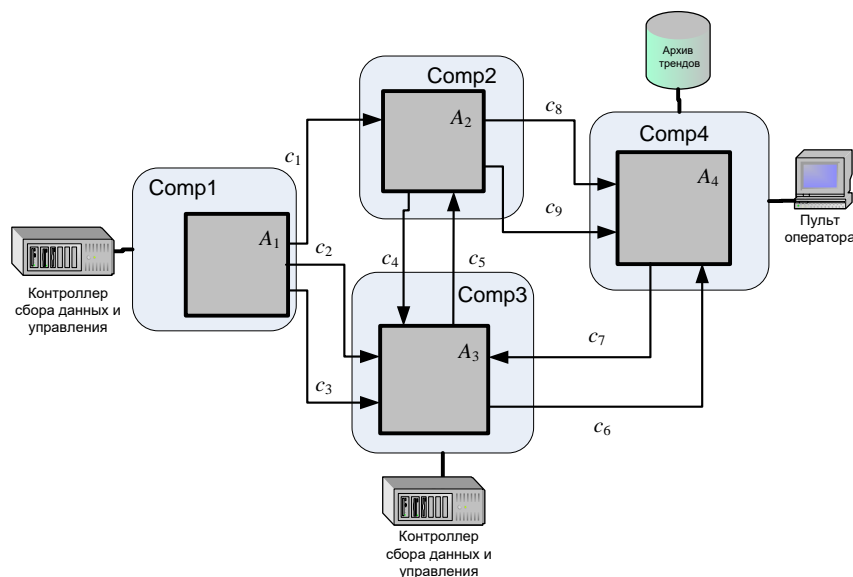


Рис. 1. Структура ПРВ на агрегатном уровне

*Модули репликации данных* (глобальной или локальной) осуществляют в БОД репликацию однородных данных либо в локальных ячейках БОД агрегата (С-репликаторы), либо в удалённых ячейках БОД разных агрегатов (R- и S-репликаторы), используя каналы передачи сообщений. *Модули терминции данных* взаимодействуют с *окружением* агрегата, обеспечивая передачу данных из ячеек БОД во внешние устройства узла ЛС. В состав БОД входит также специальный модуль - *Часы ПРВ*, которые на базе системной службы времени ОСРВ на узле ЛС формируют в ячейке *Time* значения относительного времени, используемые БОД при формировании характеристик темпоральных данных.

*Операционные модули преобразования данных* являются одновременно потребителями и поставщиками оперативных данных БОД. Они в реальном времени по часам ПРВ осуществляют параллельное и асинхронное выполнение *абстрактных операций* над оперативными данными, получаемыми из ячеек БОД, а результаты в виде темпоральных данных помещают обратно в ячейки БОД. Набор операционных модулей формально не ограничен и может эволюционировать по мере приближения прототипа ПРВ к окончательному виду приложения. Изменение состава операционных модулей в общем случае приводит к изменению количества ячеек БОД, используемых для хранения результатов обработки. Для формирования абстрактных ячеек БОД используется средство ОСРВ QNX Neutrino для создания в оперативной памяти узла ЛС *именованных областей*.

Все рассмотренные выше модули управления оперативными данными БОД и операционные модули преобразования данных формируются в ОСРВ как *процессы* с универсальной структурой, запускаемые при загрузке агрегатов



в узлы ЛС. Универсальная структура операционного модуля в виде контейнера параллельно и асинхронно выполняемых абстрактных операций над ячейками оперативной памяти БОД представлена на рисунке 3. Абстрактные операции формируются в модуле путём запуска средствами ОСРВ параллельно выполняемых *нитей* -  $O_i$ . Загруженный операционный модуль предварительно формирует *карту отображений* в модуль именованных ячеек БОД, требуемых для выполнения абстрактных операций. Запущенные нити осуществляют периодический вызов *процедур обработки данных* -  $Pr_i$ , в одном из трёх режимов: *master* – ожидание истечения заданного интервала времени, *slave* – ожидание модификации заданных ячеек БОД, *sporadic* – ожидание завершения выполнения процедуры  $Pr_i$ .

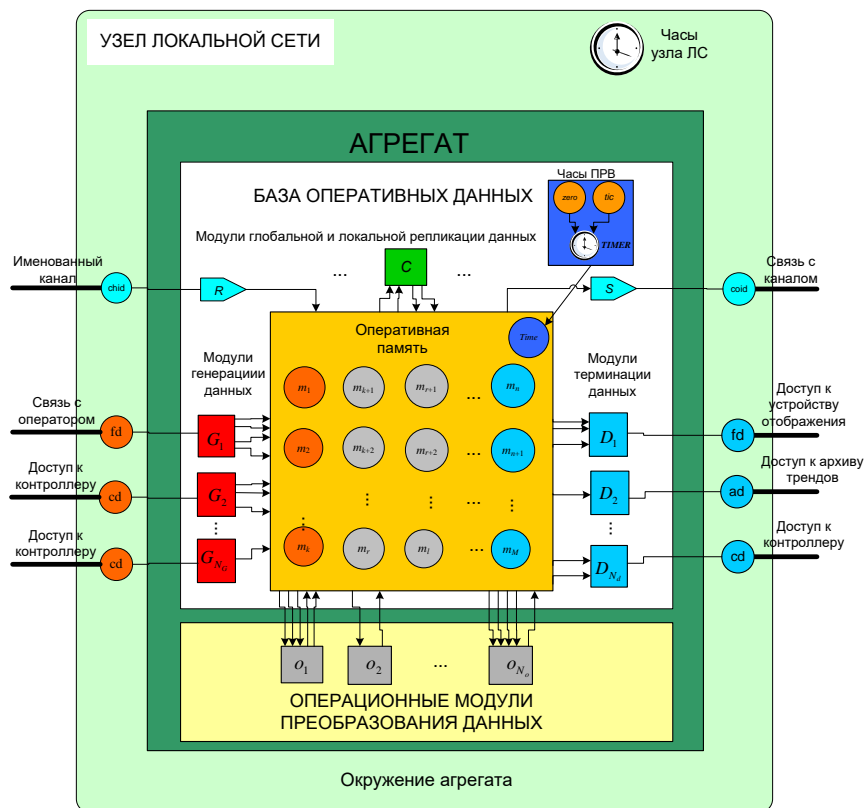


Рис. 2. Структура агрегата на процессном уровне

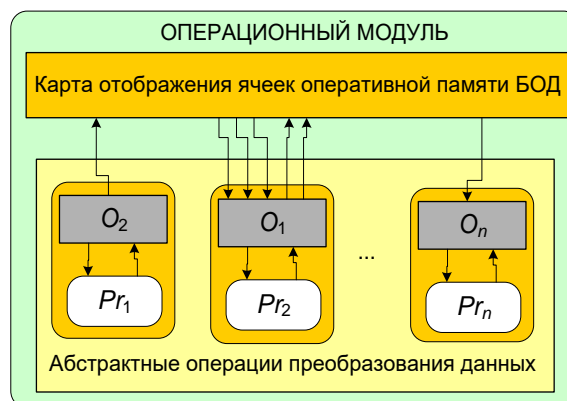


Рис. 3. Структура операционного модуля на нитевом уровне



Процедуры  $Pr_i$  составляют *процедурный уровень* ПРВ. В начале прототипирования в качестве процедур обработки данных используются специальные процедуры-«заглушки» имитирующие обработку данных. По мере развития натурального макета ПРВ они заменяются реальными процедурами.

Использование в БОД темпоральных данных позволяет на всех стадиях прототипирования осуществлять РВ-верификацию натурального макета ПРВ [4] с целью выявления так называемых *темпоральных прецедентов* нарушения временной целостности данных, используемых процедурами обработки данных, и поиска путей их устранения за счет модификации и эквивалентных преобразований структуры ПРВ на различных уровнях.

В настоящее время разработаны экспериментальные средства спецификации прототипа ПРВ на всех четырёх уровнях и загрузки натурального макета ПРВ в локальную сеть под управлением ОСРВ QNX Neutrino, которые проходят апробацию на различных примерах абстрактных распределённых ПРВ с целью проверки эффективности принятых технологических решений.

### Литература

1. Баландин, А.В., Николаев, А.В. Метод структуризации и РВ-верификации приложений реального времени для систем промышленной автоматизации // Надежность и качество. - Труды международного симпозиума. – Пенза: Изд-во Пенз. гос. ун-та. 2003. – С.378-380.
2. Баландин, А.В., Поточные диаграммы асинхронных темпоральных вычислений для моделирования и РВ-верификации приложений реального времени // Информационные технологии и нанотехнологии (ИТНТ-2016). – Сб. трудов международной конференции. – Самара: Изд-во СГАУ. 2016. – С.919-926.
3. Кёртен, Р. Введение в QNX Neutrino 2. Руководство для разработчиков приложений реального времени. – СПб.: БХВ-Петербург, 2005. – 400 с.: ил.
4. Баландин, А.В. Модель параллельных и асинхронных темпоральных вычислений с автовалидацией // Перспективные информационные технологии (ПИТ 2015). – Том 2: труды международной научно-технической конференции / под ред. С.А. Прохорова. - Самара: Изд-во Самарского научного центра РАН. 2015. – С.3-7.